

**RANCANG BANGUN PAPAN SKOR OLAHRAGA BERBASIS
MIKROKONTROLER ATMEGA 8535 DENGAN INPUTAN
KEYBOARD KOMPUTER**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Elektro Pada
Jurusan Teknik Elektro**

Oleh:

**SUPRIANTO
10355023119**



**JURUSAN TEKNIK ELEKTRO
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
2011**

**RANCANG BANGUN PAPAN SKOR OLAHRAGA BERBASIS
MIKROKONTROLER ATMEGA 8535 DENGAN INPUTAN
KEYBOARD KOMPUTER**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Elektro Pada
Jurusan Teknik Elektro**

Oleh:

SUPRIANTO
10355023119



**JURUSAN TEKNIK ELEKTRO
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
2011**

**RANCANG BANGUN PAPAN SKOR OLAHRAGA BERBASIS
MIKROKONTROLER ATMEGA 8535 DENGAN INPUTAN
KEYBOARD KOMPUTER**

SUPRIANTO

10355023119

Tanggal Sidang : 10 Juni 2011

Periode Wisuda : 14 Juli 2011

Jurusan Teknik Elektro
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Jl. HR. Soebrantas No. 155 Pekanbaru

ABSTRAK

Papan skor merupakan salah satu alat bantu dalam bidang olahraga yang berfungsi untuk mengumumkan dan mencatat hasil suatu pertandingan sehingga para penonton dan atlet dapat mengetahui hasil skor pertandingan. Papan skor ini menggunakan mikrokontroler ATmega 8535 sebagai penyimpan dan proses program data kemudian ditampilkan kerangkaian *seven-segment*, *keyboard* komputer sebagai *input* data agar memudahkan kita dalam penggunaan alat serta dapat di aplikasikan pada beberapa macam cabang olahraga seperti Volley, Sepak Takraw, futsal serta Bulutangkis. Papan skor ini juga dapat digunakan pada tempat olahraga yang permanen seperti stadion atau tempat olahraga yang semi permanen seperti lapangan terbuka.

Kata kunci : *Keyboard* Komputer, Mikrokontroler ATmega 8535, *Seven Segment*

***SPORTS SCORE BOARD DESIGN BASED ATMEGA 8535
MICROCONTROLLER WITH COMPUTER KEYBOARD INPUT***

SUPRIANTO

10355023119

*Date of final exam : 10 June 2011
Graduation Ceremony Period : 14 July 2011*

*Electrical Engineering Department
Faculty of Science and Technology
State Islamic University of Sultan Sharif Kasim Riau
HR. Soebrantas Street No. 155 Pekanbaru*

ABSTRACT

The scoreboard is one tool help in the field of sports that serves to announce and record the outcome of a match so that the spectators and athletes can find out the results of the match score. This score board using the microcontroller ATmega 8535 as a store and process data program is displayed to seven-segment, the computer keyboard as input data to help us in the use of tools and can be applied in several kinds of sports like Volley, Football Takraw, Futsal and Badminton. This scoreboard can also be used in the place of a permanent sports like stadium or a semi-permanent sporting venues such as the open field.

Keywords : Computer Keyboard, Microcontroller ATmega8535, Seven Segment

DAFTAR ISI

	Halaman
COVER	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK INTELEKTUAL.....	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii

BAB I PENDAHULUAN

1.1	Latar Belakang Masalah.....	I-1
1.2	Rumusan Masalah	I-2
1.3	Tujuan	I-2
1.4	Batasan Masalah.....	I-2
1.5	Metode Penelitian.....	I-2
1.6	Sistematika Penulisan	I-3

BAB II LANDASAN TEORI

2.1	Mikrokontroler ATmega 8535.....	II-1
2.1.1	Port Sebagai <i>Input Output</i>	II-3
2.1.2	EEPROM	II-3
2.1.3	SRAM	II-4
2.2	LED (Light Emitting Diode).....	II-4
2.3	Catu Daya.....	II-4
2.4	<i>Keyboard</i> Komputer.....	II-5

2.5	IC (<i>Integrated Circuit</i>) 4094	II-6
2.6	Bascom Avr (<i>Alf and Vegard's Risc processor</i>)	II-7

BAB III PERANCANGAN SISTEM

3.1	Diagram Blok Sistem	III-1
3.2	Perancangan Rangkaian Elektronika	III-2
3.2.1	Catu Daya	III-2
3.2.2	Rangkaian <i>Port</i> USB	III-2
3.2.3	Rangkaian Seven Segment	III-3
3.3	Perancangan Perangkat Lunak Sistem	III-5
3.4	Kode ASCII	III-7

BAB IV HASIL DAN ANALISIS

4.1	Hasil	IV-1
4.1.1	Tampilan Simulasi <i>Seven Segment</i>	IV-1
4.1.2	Pengujian Rangkaian <i>Seven Segment</i>	IV-4
4.2	Analisis	IV-15
4.2.1	Rangkaian Penstabil Sinyal	IV-15
4.2.2	Proses Pengiriman Data Rangkaian <i>Seven Segment</i>	IV-17

BAB V KESIMPULAN DAN SARAN

5.1	Kesimpulan	V-1
5.2	Saran	V-1

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam dunia olahraga papan skor memiliki fungsi penting karena merupakan alat bantu untuk mengumumkan dan mencatat hasil suatu pertandingan sehingga para penonton dan atlet dapat mengetahui hasil skor pertandingan. Banyak cara untuk mendapatkan hasil penghitungan skor pertandingan mulai dari menulis pada potongan kertas karton atau papan triplex, menyebutkan pada alat pengeras suara atau mikrofon dan lain-lain.

Pada perancangan sebelumnya papan skor olahraga ini dirancang (Budi, Mayan 2008), (Wahyudi, 2010), dan (Suryo, kabut 2008) menggunakan mikrokontroler AT89S52 dan AT89C51 dengan tampilan *seven segment* dan papan skor tersebut hanya dapat diaplikasikan pada satu jenis cabang olahraga saja. Pengembangan papan skor nantinya akan diaplikasikan pada cabang olahraga Volley, Sepak Takraw, futsal serta Bulutangkis. Mikrokontroler yang dipakai jenis ATmega 8535 karena mikrokontroler ini memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 512 Byte, SRAM (*Static Random Access Memory*) 512 Byte, sedangkan pada mikrokontroler AT89S52 memiliki SRAM (*Static Random Access Memory*) 256 Byte dan AT89C51 memiliki SRAM (*Static Random Access Memory*) 128 Byte serta kedua mikrokontroler ini tidak memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) internal. Teknologi pada judul papan skor sebelumnya menggunakan *input* data berupa tombol *switch* untuk menambah jumlah hasil skor pertandingan sehingga fitur yang ditampilkan hanya berupa angka saja sedangkan nama tim yang bertanding ditampilkan secara manual.

Berdasarkan latar belakang diatas, penulis tertarik untuk **rancang bangun papan skor olahraga berbasis Mikrokontroler ATmega 8535 dengan inputan keyboard komputer** dengan tujuan untuk mempermudah dan menambah fitur sehingga operator dapat menampilkan skor pertandingan berserta nama tim yang sedang bertanding selain itu papan skor ini diciptakan dengan bobot yang cukup

ringan sehingga dapat diaplikasikan pada tempat olahraga yang permanen seperti stadion atau tempat olahraga yang semi permanen seperti lapangan terbuka.

1.2. Rumusan Masalah

Bagaimana merancang dan membuat papan skor olahraga berbasis mikrokontroler ATmega 8535.

1.3. Tujuan

Tujuan pembuatan tugas akhir ini adalah sebagai berikut :

- Mampu membuat papan skor olahraga berbasis mikrokontroler ATmega 8535.
- Membuat papan skor olahraga dengan kemampuan menampilkan *output* berupa karakter huruf dan angka.

1.4. Batasan Masalah

Permasalahan pada proposal tugas akhir ini dibatasi pada rancang bangun papan skor olahraga :

- Menggunakan Mikrokontroler ATmega 8535.
- Menggunakan LED untuk rangkaian *seven segment* sebagai *output*.
- Menggunakan *keyboard* komputer sebagai *input*.
- Menggunakan bahasa pemrograman Visual Basic.
- Menggunakan *software circuit maker 6 pro* untuk simulasi rangkaian *seven segment*.
- Menggunakan *software proteus 7 professional* untuk perancangan rangkaian elektronik.

1.5. Metode Penelitian

Metode yang dipilih dalam **rancang bangun papan skor olahraga berbasis Mikrokontroler ATmega 8535 dengan inputan *keyboard* komputer** ini adalah metode eksperimen dan studi pustaka. Perancangan dan pembuatan

proposal tugas akhir ini memerlukan langkah-langkah penyelesaian sebagai berikut :

- Studi Pustaka
Metode studi pustaka dilakukan melalui buku acuan atau *textbook*, data *sheet*, artikel, *internet resource* yang berhubungan dengan proposal tugas akhir.
- Perancangan dan pembuatan
Merancang dan membuat peralatan serta sistem yang dibutuhkan berupa perangkat keras dan perangkat lunak.
- Hasil dan analisis
Mengintegrasikan sistem antara perangkat keras dengan perangkat lunak, kemudian dilakukan pengujian dan analisis terhadap hasil yang didapatkan.
- Penulisan Laporan
Membuat laporan yang berisikan tentang kegiatan-kegiatan dari awal pembuatan hingga pengujian tugas akhir.

1.6. Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini menguraikan secara umum dan singkat mengenai Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan, Metodologi penelitian dan Sistematika Penulisan yang diterapkan.

BAB II LANDASAN TEORI

Bab ini berisi mengenai teori dan metode yang mendukung dalam pembuatan proposal Tugas Akhir ini.

BAB III PERANCANGAN SISTEM

Bab ini berisi mengenai perancangan dan pembuatan alat yang akan dibangun.

BAB IV HASIL DAN ANALISIS

Bab ini berisi mengenai proses pengujian dan menganalisa hasil dari alat yang dibuat.

BAB V KESIMPULAN DAN SARAN

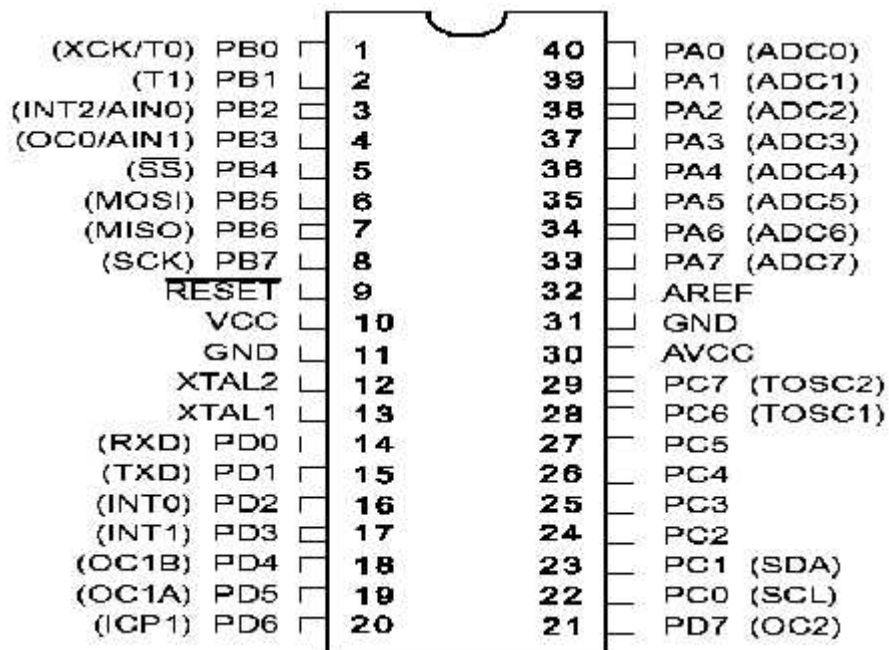
Bab ini berisi mengenai kesimpulan dan saran hasil dari pengujian dan analisa dari alat yang dibuat.

BAB II

LANDASAN TEORI

2.1. Mikrokontroler ATmega 8535

ATMega 8535 ialah IC mikrokontroler 8 bit CMOS daya rendah berbasis AVR (*Alf and Vegard's Risc processor*) dengan arsitektur RISC (*Reduce Instruction Set Computing*) yang ditingkatkan. Hampir semua instruksi dieksekusi dalam 1 siklus clock, berbeda dengan instruksi MCS 51 yang membutuhkan 12 siklus clock. Hal ini dikarenakan MCS 51 memiliki teknologi CISC (*Complex Instruction Set Computing*). Didalam AVR CPU terdapat SRAM 512 byte, *Stack Pointer*, Memori program, dan Program counter. AVR memiliki *feature* EEPROM 512 Byte, *Timer/ Counter*, ADC *internal*, Komunikasi USART, Mode SPI. Konfigurasi ATMega 8535 yang memiliki 40 pin DIP (*Dual in line package*) seperti pada Gambar 1.



Gambar 2.1. Pin ATmega 8535 (Lingga Wardana, 2006)

Adapun fungsi umum Pin ATmega 8535 adalah sebagai berikut :

a. VCC

Merupakan pin yang berfungsi sebagai masukan catu daya.

b. GND (*ground*)

c. Port A (PA0-PA7)

Port A selain berfungsi sebagai Port I/O 8 bit, Port A juga dapat digunakan sebagai masukan *Analog to Digital Converter*.

d. Port B (PB0-PB7)

Port B adalah suatu Port I/O 8-bit dua arah dengan fungsi khusus sebagai *timer/Counter*, komparator Analog dan SPI (*Serial peripheral interface*).

Tabel 2.1 Keterangan fungsi khusus port pin PB0-PB7. (Iswanto, 2008)

PORT PIN	FUNGSI KHUSUS
PB0	T0 = <i>timer/counter 0 external counter input</i>
PB1	T1 = <i>timer/counter 1 external counter input</i>
PB2	AIN0 = <i>analog comparator positive input</i>
PB3	AIN1 = <i>analog comparator negative input</i>
PB4	SS = <i>SPI slave select input</i>
PB5	MOSI = <i>SPI bus master output/slave input</i>
PB6	MISO = <i>SPI bus master input/slave output</i>
PB7	SCK = <i>SPI bus serial clock</i>

d. Port C (PC0-PC7)

Port C adalah suatu Port I/O 8-bit dua arah dan pin yang mempunyai fungsi khusus, yaitu komparator analog dan *timer Oscillator*

e. Port D (PD0-PD7)

Port D adalah suatu Port I/O 8-bit dua arah dan pin fungsi khusus yaitu komparator analog dan *interrupt eksternal* serta komunikasi serial

Tabel 2.2 Keterangan fungsi khusus *port* pin PD0-PD7. (Iswanto, 2008)

PORT PIN	FUNGSI KHUSUS
PD0	RDX (UART <i>input line</i>)
PD1	TDX (UART <i>output line</i>)
PD2	INT0 (<i>external interrupt 0 input</i>)
PD3	INT1 (<i>external interrupt 1 input</i>)
PD4	OCIB (<i>time/counter 1 output compare B match output</i>)
PD5	OCIA (<i>time/counter 1 output compare A match output</i>)
PD6	ICP (<i>timer/Counter 1 input capture pin</i>)
PD7	OC2 (<i>timer/Counter 2 output compare match output</i>)

- f. RESET merupakan kaki pin yang berfungsi sebagai mereset mikrokontroler.
- g. XTAL1 adalah masukan ke *inverting oscillator amplifier* dan *input* ke *internal clock operating circuit*.
- h. XTAL2 merupakan *output* dari *inverting oscillator amplifier*.
- i. AVCC adalah kaki pin masukan untuk tegangan ADC (*Analog to Digital Converter*).
- j. AREF adalah kaki pin masukan tegangan referensi untuk ADC (*Analog to Digital Converter*).

2.1.1 Port Sebagai *Input Output*

Pada IC mikrokontroler ATmega 8535 telah memiliki empat buah *port* yaitu *Port A*, *Port B*, *Port C*, *Port D*. Keempat *port* tersebut akan berfungsi sebagai jalur *output* atau *input* data I/O.

2.1.2 EEPROM (*Electrically Erasable Programmable Read Only Memory*)

EEPROM (*Electrically Erasable Programmable Read Only Memory*) ialah salah satu tipe memori AVR (*Atmel and Vegard's Risc processor*). EEPROM akan tetap dapat menyimpan data meskipun catu daya telah dimatikan dan dapat diubah saat program berjalan. Pada ATmega 8535 EEPROM (*Electrically Erasable*

Programmable Read Only Memory) yang dimiliki sebesar 512 *byte* dan mempunyai masa daya tahan 100.000 kali *write / erase* program.

2.1.3 SRAM (*Static Random Access Memory*)

SRAM (*Static Random Access Memory*) merupakan salah satu tipe memori yang hanya dapat menyimpan data sementara, ATmega 8535 memiliki SRAM (*Static Random Access Memory*) sebesar 512 *byte*.

2.2. LED (*Light Emitting Diode*)

LED (*Light Emitting Diode*) merupakan salah satu komponen elektronik yang sering kita jumpai pada kehidupan sehari – hari, contohnya pada lampu emergency, senter dan mainan anak dll. LED (*Light Emitting Diode*) adalah piranti yang dapat mengeluarkan atau memancar cahaya bila diberi tegangan. Seiring dengan perkembangan zaman pada saat ini LED (*Light Emitting Diode*) banyak mengalami perkembangan salah satunya ialah penggunaan LED pada layar televisi dan laptop.



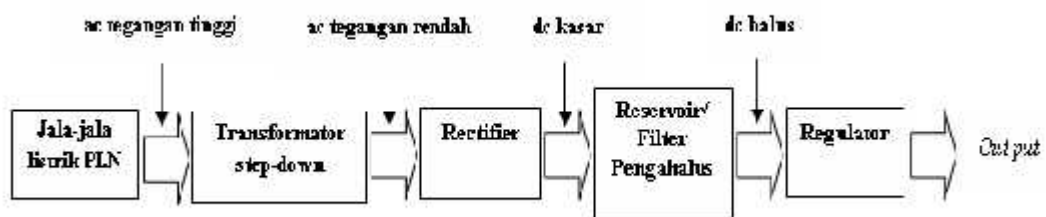
Gambar 2.2. LED (*Light Emitting Diode*)

2.3. Catu Daya

Dalam sistem elektronik, hampir semua rangkaian elektronik membutuhkan sumber tegangan DC (*direct current*) yang teratur dengan besar 5 V- 30 V. Dalam

beberapa kasus, pencatutan ini dapat dilakukan secara langsung oleh baterai (misalnya 6 V, 9 V, 12 V) namun dalam banyak kasus lainnya akan lebih menguntungkan apabila di gunakan sumber AC (*alternatting current*) standar, yaitu penghematan tanpa harus membeli baterai secara terus-menerus.

Diagram blok catu daya DC diperlihatkan pada Gambar 2.3 Karena input sumbernya memiliki tegangan yang relatif tinggi, maka digunakanlah sebuah *ransformator step-down* dengan *ratio* lilitan yang sesuai untuk mengkonversi tegangan ini ke tegangan rendah. *Output* ac dari sisi sekunder *transformator* kemudian disearahkan dengan menggunakan dioda-dioda penyearah (*rectifier*) silikon konvensional untuk menghasilkan *output* yang masih kasar (kadang kala disebut sebagai DC berdenyut). *Output* ini kemudian di haluskan dan kemudian di *filter* sebelum di salurkan ke *input* rangkaian.



Gambar 2.3. Diagram Blok Catu Daya DC

2.4. Keyboard Komputer

Keyboard merupakan alat penghubung antara manusia dengan komputer dimana fungsinya ialah memberikan perintah berupa huruf atau angka kemudian menampilkan ke layar monitor yang sebelumnya perintah tersebut diolah secara elektronis oleh CPU (*Central Processing Unit*).

Pada *keyboard* komputer memiliki *port* penghubung bernama USB (*Universal Serial Bus*) yang berfungsi untuk menghubungkan *keyboard* tersebut ke perangkat lain seperti komputer dan lain-lain.



Gambar 2.4. *Keyboard Komputer*

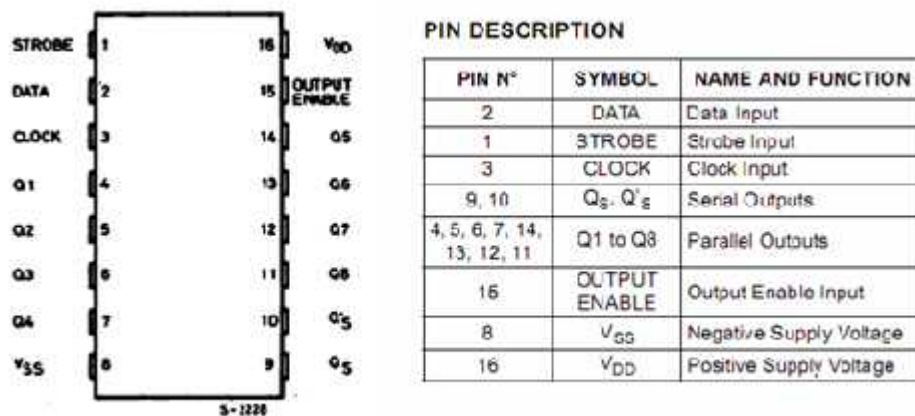


Gambar 2.5. USB (*Universal Serial Bus*)

Sumber : (http://en.wikipedia.org/wiki/Universal_Serial_Bus. diakses tanggal 14 maret 2011)

2.5. IC (*Integrated Circuit*) 4094

IC (*Integrated Circuit*) merupakan suatu rangkaian elektronik yang dikemas menjadi satu kemasan yang kecil. Salah satu IC yang sering kita jumpai ialah IC 4094 memiliki enam belas buah kaki, berikut merupakan bentuk dari IC 4094.



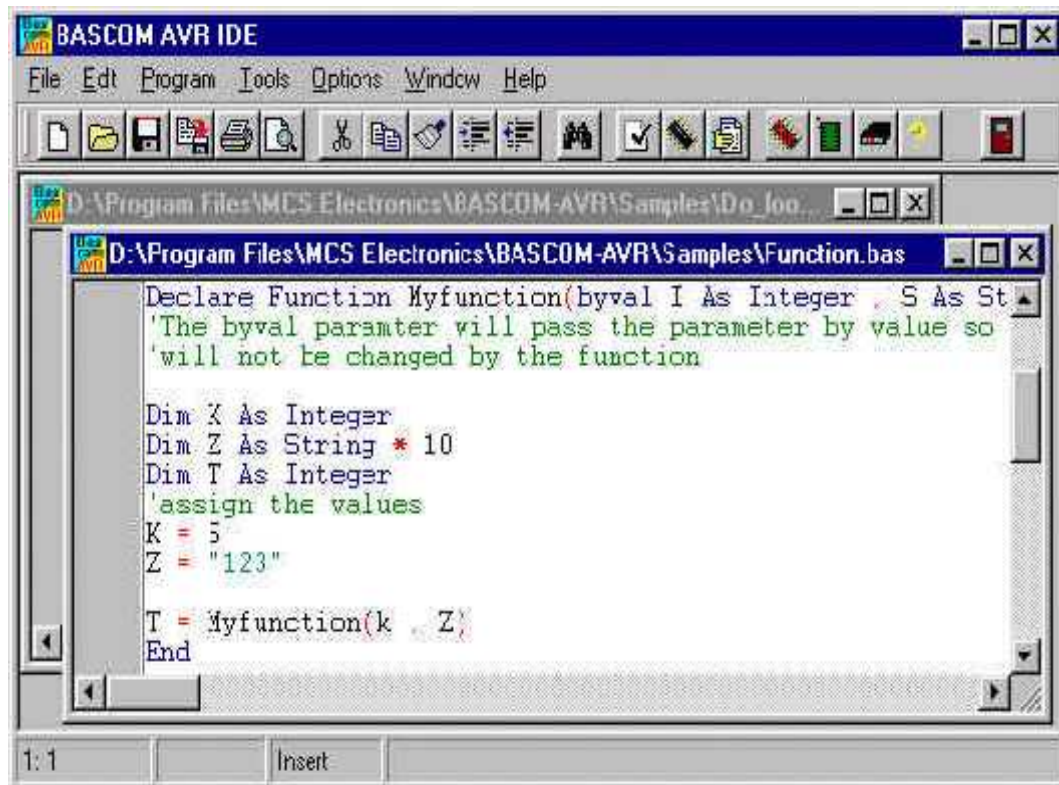
Gambar 2.6 IC (*Integrated Circuit*) 4094

2.6. BASCOM AVR (*Alf and Vegard's Risc processor*)

Mikrokontroler merupakan *chip* cerdas yang menjadi tren dalam pengendalian dan otomatisasi, terutama di kalangan mahasiswa. Dengan banyak jenis keluarga, kapasitas memori, dan berbagai fitur, mikrokontroler menjadi pilihan dalam aplikasi prosesor mini untuk pengendalian skala kecil, oleh karena itu mikrokontroler membutuhkan sebuah perintah atau yang biasa disebut bahasa pemrograman.

BASCOM AVR adalah suatu *software* yang digunakan untuk mengubah perintah program yang kita buat menjadi suatu file yang dapat dibaca sekaligus dijalankan oleh micro. Dalam hal ini program diubah menjadi **FILE.HEX**. **BASCOM** dikembangkan oleh **MCS Electronics**, dan merupakan **BASIC compiler**. Program yang dibuat dalam bahasa BASIC, akan dikompilasi menjadi *machine code*, untuk kemudian dimasukkan kedalam mikrokontroler melalui sebuah programmer.

Berikut adalah contoh tampilan program BASCOM AVR



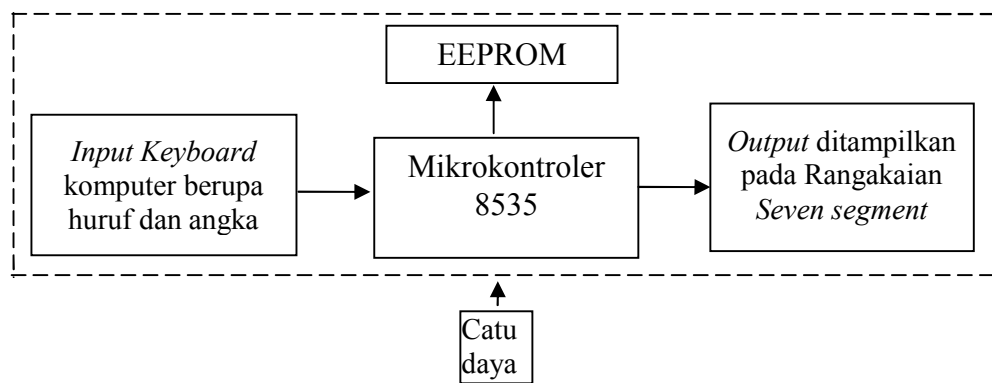
Gambar 2.7 Tampilan program BASCOM AVR

BAB III

PERANCANGAN SISTEM

3.1. Diagram Blok Sistem

Pada perancangan sistem, diagram blok sistem sangatlah penting untuk kejelasan bentuk dan alur sistem yang dirancang sehingga dapat mudah dimengerti tiap bagian-bagian dari alat tersebut fungsi Gambar 3.1. menunjukkan diagram blok sistem papan skor dirancang.



Gambar 3.1 Blok Diagram Sistem

Fungsi masing-masing blok diagram sistem Gambar 3.1 di atas adalah sebagai berikut :

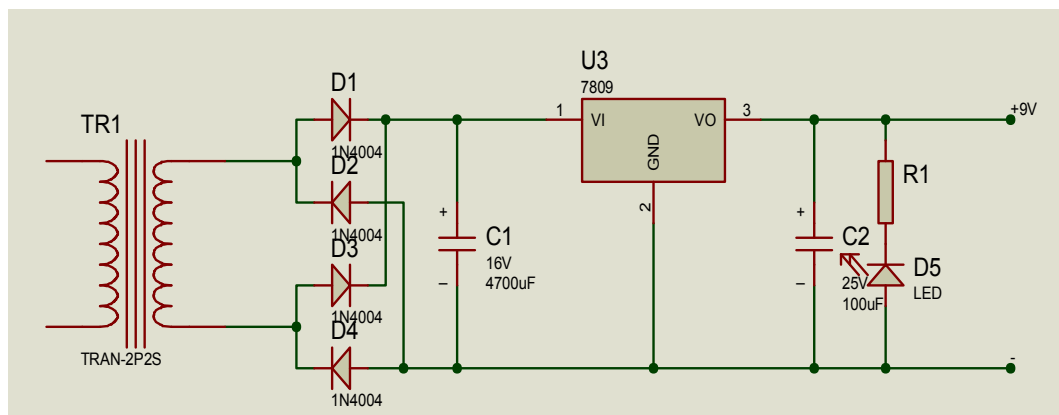
1. Keyboard, berfungsi untuk menginput data berupa huruf dan angka dan mengirimkan ke mikrokontroler.
2. Mikrokontroler, berfungsi sebagai pengolah data yang dikirim dari keyboard.
3. EEPROM, berfungsi sebagai penyimpan data.
4. *Seven segment*, merupakan *output* akhir dari blok diagram sistem dan berfungsi untuk menampilkan data berupa huruf dan angka.
5. Catu daya, berfungsi sebagai pemberi masukan tegangan pada seluruh rangkaian.

3.2. Perancangan Rangkaian Elektronika

Perancangan perangkat keras ini terdiri dari beberapa kit komponen antara lain adalah mikrokontroler Atmega 8535, *keyboard* komputer, rangkaian *seven segment* dan rangkaian catu daya. Bagian-bagian ini mempunyai fungsi yang berbeda-beda sehingga membentuk satu kesatuan sistem perangkat yang utuh.

3.2.1 Catu Daya

Pemberian catu daya tegangan pada rangkaian elektronika pada perancangan ini menggunakan adaptor DC. Tegangan keluaran yang diperlukan adalah 9V dengan menggunakan regulator dari IC 78L09.

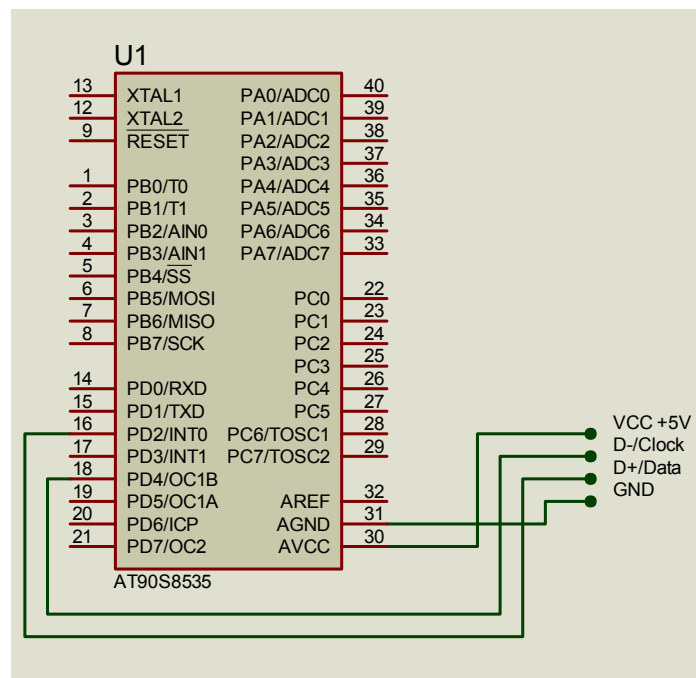


Gambar 3.2 Rangkaian catu daya

Gambar 3.2 memperlihatkan rangkaian catu daya dengan beberapa komponen antara lain transformator *step-down*, dioda, kapasitor, IC, resistor dan LED indikator. Tegangan keluaran 9V digunakan untuk catu daya Mikrokontroler.

3.2.2 Rangkaian Port USB (*Universal Serial Bus*)

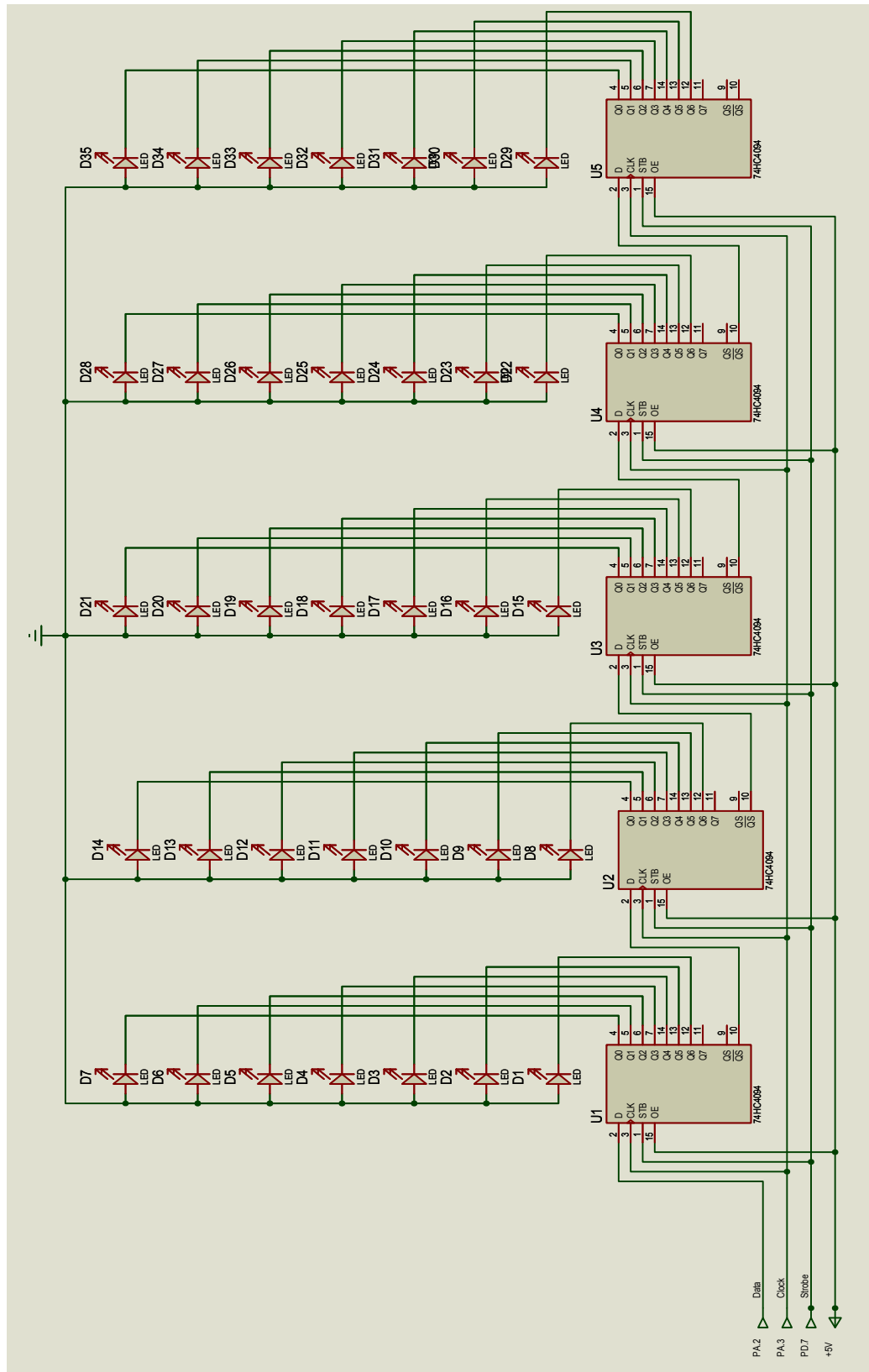
Rangkaian *port* USB berfungsi untuk menghubungkan perangkat *keyboard* dengan mikrokontroler sehingga pada saat alat ingin digunakan dapat dihubungkan dan ketika tidak digunakan dapat dilepas kembali, selain itu *port* ini merupakan jalur *input* yang dilewati data dan *clock*. *Port* USB terdiri dari empat kaki yaitu: VCC, D-/clock, D+/data, *ground*.



Gambar 3.3 Rangkaian port USB (*Universal Serial Bus*)

3.2.3 Rangkaian *Seven Segment*

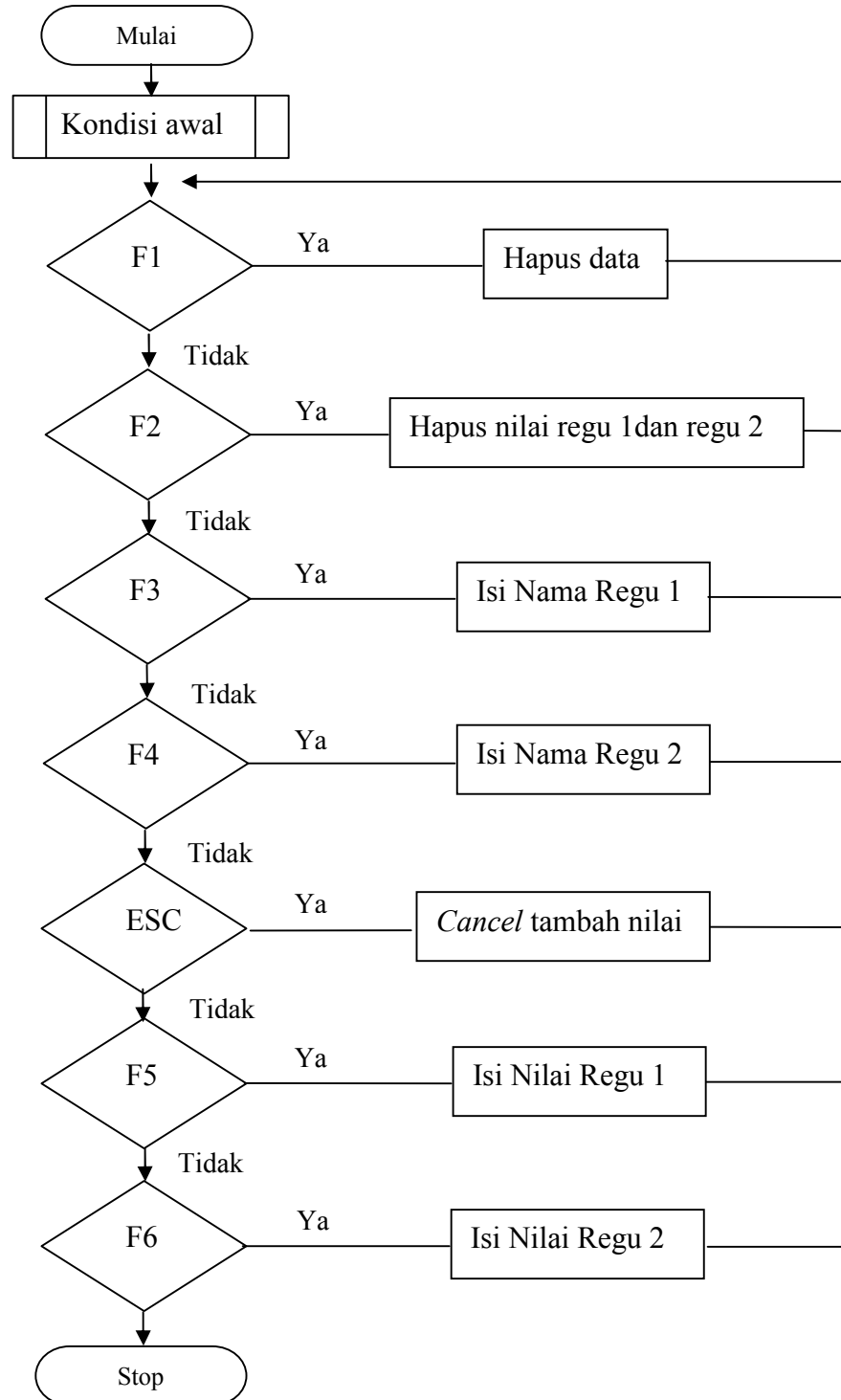
Pada rangkaian *seven segment* terdapat beberapa komponen dalam satu keping papan pcb diantaranya ialah LED 2V sebanyak 35 buah berfungsi sebagai *display*, serta menggunakan IC 4094 (16 kaki) sebanyak 5 buah yang berfungsi untuk mengatur nyala LED sesuai dengan program yang dibuat, memiliki delapan register dengan penguci penyimpanan, merubah *input* serial kedelapan bit paralel, data akan disimpan pada register penyimpanan bila *strobe* berlogika satu, memiliki pin Qs dan Q's yang berfungsi untuk keperluan penyambungan beberapa IC yang sama secara serial, data akan disimpan bila OE (*output enable*) sama dengan satu, selain itu IC 4094 juga berfungsi sebagai pintu gerbang data yang di kirim dari mikrokontroler agar pin *output* dapat dihemat.



Gambar 3.4 Rangkaian *seven segment*

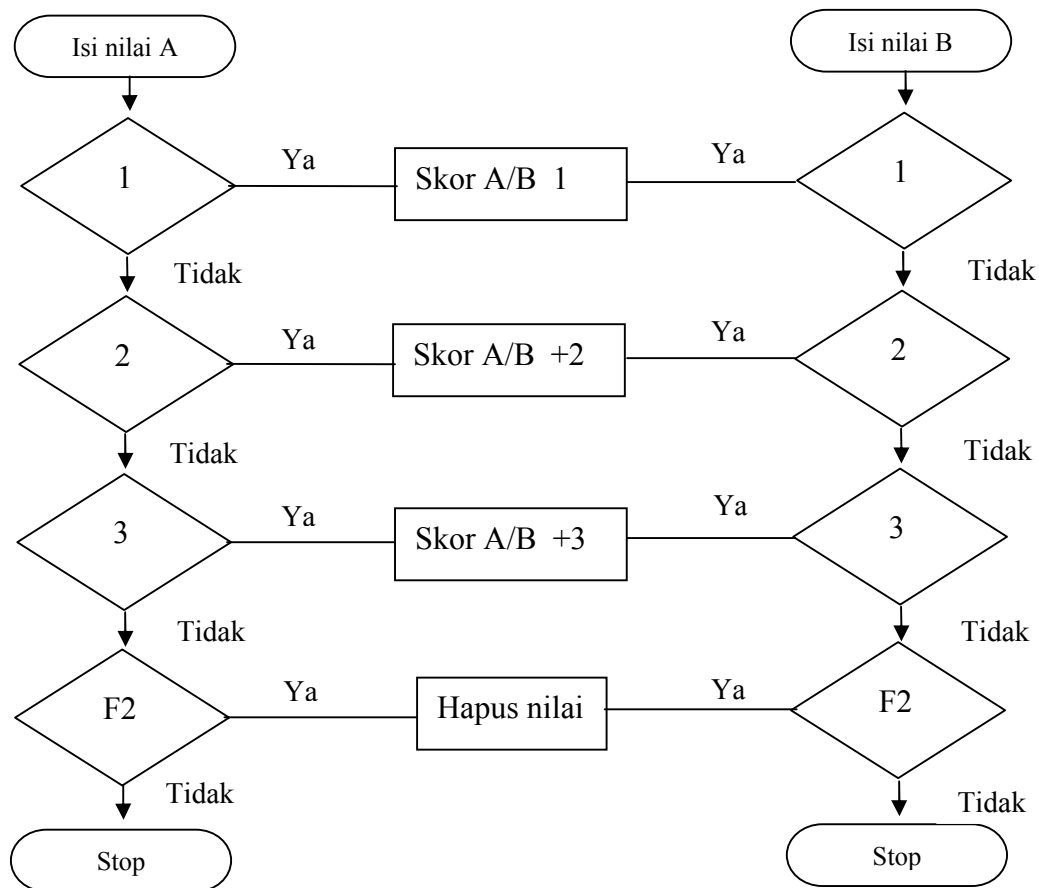
3.3. Perancangan Perangkat Lunak Sistem

Perancangan pemrograman komputer diawali dengan perancangan diagram alir untuk mempermudah dalam menunjukkan alur pemrograman



Gambar 3.5 Flow chart perancangan sistem

Pada gambar di bawah menunjukan sub *flow chart* perancangan sistem, fungsi dari tombol F5 dan F6 adalah untuk menambah jumlah skor hasil pertandingan dari regu 1 dan regu 2 hingga mencapai skor maksimal pertandingan. Skor maksimal yang mampu tampilkan pada alat ini hanya sampai angka 90.



Gambar 3.6 Sub *Flow chart* perancangan sistem

Agar memudahkan kita dalam pengoperasian alat berikut beberapa fungsi tombol *keyboard* komputer yang dipakai :

F1 = hapus data

F2 = hapus nilai regu 1 dan regu 2

F3 = *input* nama tim 1

F4 = *input* nama tim 2

ESC = *Cancel* tambah nilai

F5 = *input* nilai regu 1

F6 = *input* nilai regu 2

Shift + huruf = huruf besar

Enter = spasi

3.4. Kode ASCII (*American Standard Code for Information Interchange*)

Pada pemrograman papan skor, mikrokontroler membutuhkan kode ASCII (*American Standard Code for Information Interchange*) agar dapat mengenali perangkat *keyboard* sehingga *keyboard* bisa memberikan perintah ke mikrokontroler dan menampilkan ke *display*. Cara kerja mikrokontroler mengenali *keyboard* ialah dengan menscan setiap persekian detik dan ini akan terus berulang sampai salah satu tombol *keyboard* ditekan. Berikut data kode ASCII yang digunakan:

```
Data ""
Data "F9" , "" , "F5" , "F3" , "F1" , "F2" , "F12" , "" , "F10" ,
"F8"
Data "F6" , "F4" , "TAB" , "`" , "" , "" , "LAlt" , "LSh" , "" ,
"LCtr" , "q"
Data "1" , "" , "" , "" , "z" , "s" , "a" , "w" , "2" , ""
Data "" , "c" , "x" , "d" , "e" , "4" , "3" , "" , "" , ""
Data "v" , "f" , "t" , "r" , "5" , "" , "" , "n" , "b" , "h"
Data "g" , "y" , "6" , "" , "" , "" , "m" , "j" , "u" , "7"
Data "8" , "" , "" , "" , "k" , "i" , "o" , "0" , "9" , ""
Data "" , "." , "/" , "l" , ";" , "p" , "-" , "" , "" , ""
Data "'" , "" , "[" , "=" , "" , "" , "Caps" , "RSht" , "Enter" ,
"]"
Data "" , "" , "" , "" , "" , "" , "" , "" , "" , ""
Data "BkSp" , "" , "" , "1" , "" , "4" , "7" , "" , "" , ""
Data "0" , "." , "2" , "5" , "6" , "8" , "ESC" , "NumLck" , "F11" ,
"+"
Data "3" , "-" , "*" , "9" , "ScrLck" , "" , "" , "" , "" , "F7"
')
Data 0
Data 137 , 0 , 133 , 131 , 129 , 130 , 140 , 0 , 138 , 136
Data 134 , 132 , 9 , 96 , 0 , 0 , 0 , 0 , 0 , 113
Data 49 , 0 , 0 , 0 , 122 , 115 , 97 , 119 , 50 , 0
Data 0 , 99 , 120 , 100 , 101 , 52 , 51 , 0 , 0 , 32
Data 118 , 102 , 116 , 114 , 53 , 0 , 0 , 110 , 98 , 104
Data 103 , 121 , 54 , 0 , 0 , 0 , 109 , 106 , 117 , 55
Data 56 , 0 , 0 , 44 , 107 , 105 , 111 , 48 , 57 , 0
Data 0 , 46 , 47 , 108 , 59 , 112 , 45 , 0 , 0 , 0
Data 39 , 0 , 91 , 61 , 0 , 0 , 0 , 0 , 13 , 93
Data 0 , 92 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0
Data 8 , 0 , 0 , 49 , 0 , 52 , 55 , 0 , 0 , 0
Data 48 , 46 , 50 , 53 , 54 , 56 , 27 , 0 , 139 , 43
Data 51 , 45 , 42 , 57 , 0 , 0
```

```
'dengan shift:
'(
Data ""
Data "F9" , "" , "F5" , "F3" , "F1" , "F2" , "F12" , "" , "F10" ,
"F8"
Data "F6" , "F4" , "TAB" , "~" , "" , "" , "LAlt" , "LSh" , "" ,
"LCtr" , "Q"
Data "!" , "" , "" , "" , "Z" , "S" , "A" , "W" , "@" , ""
Data "" , "C" , "X" , "D" , "E" , "$" , "#" , "" , "" , ""
Data "V" , "F" , "T" , "R" , "%" , "" , "" , "N" , "B" , "H"
Data "G" , "Y" , "^" , "" , "" , "" , "M" , "J" , "U" , "&"
Data "*" , "" , "" , "<" , "K" , "I" , "O" , ")" , "(" , ""
Data "" , ">" , "?" , "L" , ":" , "P" , "" , "" , "" , ""
Data 34 , 0 , "" , "{" , "+" , "" , "" , "" , "Caps" , "RSht" , "Enter"
, "}"
Data "" , "|" , "" , "" , "" , "" , "" , "" , "" , ""
Data "BkSp" , "" , "" , "1" , "" , "4" , "7" , "" , "" , ""
Data "0" , "." , "2" , "5" , "6" , "8" , "ESC" , "NumLck" , "F11"
, "+"
Data "3" , "-" , "*" , "9" , "ScrLck" , "" , "" , "" , "" , "F7"
')
Data 0
Data 137 , 0 , 133 , 131 , 129 , 130 , 140 , 0 , 138 , 136
Data 134 , 132 , 9 , 126 , 0 , 0 , 0 , 0 , 0 , 0 , 81
Data 33 , 0 , 0 , 0 , 90 , 83 , 65 , 87 , 64 , 0
Data 0 , 67 , 88 , 68 , 69 , 36 , 35 , 0 , 0 , 32
Data 86 , 70 , 84 , 82 , 37 , 0 , 0 , 78 , 66 , 72
Data 71 , 89 , 94 , 0 , 0 , 0 , 77 , 74 , 85 , 38
Data 42 , 0 , 0 , 60 , 75 , 73 , 79 , 41 , 40 , 0
Data 0 , 62 , 63 , 76 , 58 , 80 , 95 , 0 , 0 , 0
Data 34 , 0 , 123 , 43 , 0 , 0 , 0 , 0 , 13 , 125
Data 0 , 124 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0
Data 8 , 0 , 0 , 49 , 0 , 52 , 55 , 0 , 0 , 0
Data 48 , 46 , 50 , 53 , 54 , 56 , 27 , 0 , 139 , 43
Data 51 , 45 , 42 , 57 , 0 , 0
```

Kode ASCII merupakan kode standar internasional yang digunakan pada *keyboard* komputer dalam pertukaran informasi dari bentuk desimal atau heksadesimal menjadi karakter dan simbol, seperti yang terlihat pada tabel berikut:

Tabel 3.1 kode ASCII(American Standard Code for Information Interchange)

sumber: (<http://www.asciitable.com> diakses tanggal 26 april 2011)

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	00		NUL (null)	32	20	040	SPACE	64	40	100	@	96	60	140	z
1	01		SOH (start of heading)	33	21	041	!	65	41	101	A	97	61	141	{
2	02		STX (start of text)	34	22	042	"	66	42	102	B	98	62	142	
3	03		ETX (end of text)	35	23	043	#	67	43	103	C	99	63	143	~
4	04		EOF (end of transmission)	36	24	044	\$	68	44	104	D	100	64	144	
5	05		ENQ (enquiry)	37	25	045	%	69	45	105	E	101	65	145	
6	06		ACK (acknowledge)	38	26	046	&	70	46	106	F	102	66	146	
7	07		BEL (bell)	39	27	047	'	71	47	107	G	103	67	147	
8	08		BS (backspace)	40	28	050	(72	48	110	H	104	68	150	
9	09		HT (horizontal tab)	41	29	051)	73	49	111	I	105	69	151	
10	0A		LF (NL line feed, new line)	42	2A	052	*	74	4A	112	J	106	70	152	
11	0B		VT (vertical tab)	43	2B	053	+	75	4B	113	K	107	71	153	
12	0C		FF (NP form feed, new page)	44	2C	054	,	76	4C	114	L	108	72	154	
13	0D		CR (carriage return)	45	2D	055	-	77	4D	115	M	109	73	155	
14	0E		SO (shift out)	46	2E	056	.	78	4E	116	N	110	74	156	
15	0F		SI (shift in)	47	2F	057	/	79	4F	117	O	111	75	157	
16	10		DL (data link escape)	48	30	060	0	80	50	120	P	112	76	160	
17	11		DC1 (device control 1)	49	31	061	1	81	51	121	Q	113	77	161	
18	12		DC2 (device control 2)	50	32	062	2	82	52	122	R	114	78	162	
19	13		DC3 (device control 3)	51	33	063	3	83	53	123	S	115	79	163	
20	14		DC4 (device control 4)	52	34	064	4	84	54	124	T	116	80	164	
21	15		NAK (negative acknowledge)	53	35	065	5	85	55	125	U	117	81	165	
22	16		SYN (synchronous idle)	54	36	066	6	86	56	126	V	118	82	166	
23	17		ETB (end of trans. block)	55	37	067	7	87	57	127	W	119	83	167	
24	18		CAN (cancel)	56	38	070	8	88	58	130	X	120	84	170	
25	19		EM (end of medium)	57	39	071	9	89	59	131	Y	121	85	171	
26	1A		SUB (substitute)	58	3A	072	:	90	5A	132	Z	122	86	172	
27	1B		ESC (escape)	59	3B	073	;	91	5B	133	[123	87	173	
28	1C		FS (file separator)	60	3C	074	<	92	5C	134	\	124	88	174	
29	1D		GS (group separator)	61	3D	075	=	93	5D	135]	125	89	175	
30	1E		RS (record separator)	62	3E	076	>	94	5E	136	^	126	90	176	
31	1F		US (unit separator)	63	3F	077	?	95	5F	137	_	127	91	177	

Source: www.LookupTables.com

BAB IV

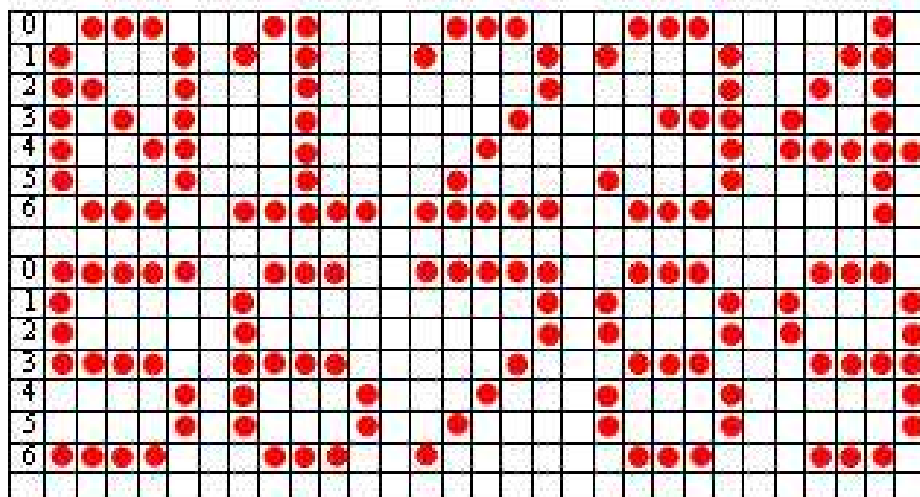
HASIL DAN ANALISIS

Setelah melakukan proses perancangan dan pembuatan sistem, tahap selanjutnya merupakan tahap pembahasan untuk mengetahui apakah hasil rancangan yang telah dibuat mencapai tujuan sesuai dengan keinginan atau belum, setelah kita melakukan pengujian dan analisa terhadap alat yang telah kita buat. Dari pembahasan ini juga nantinya kita dapat mengetahui pada bagian mana dan komponen apa yang menyebabkan alat kita tidak dapat bekerja sesuai dengan keinginan kita, Pada tahap ini juga kita dapat mengetahui akan kelebihan dan kekurangan yang dimiliki oleh alat yang telah kita buat.

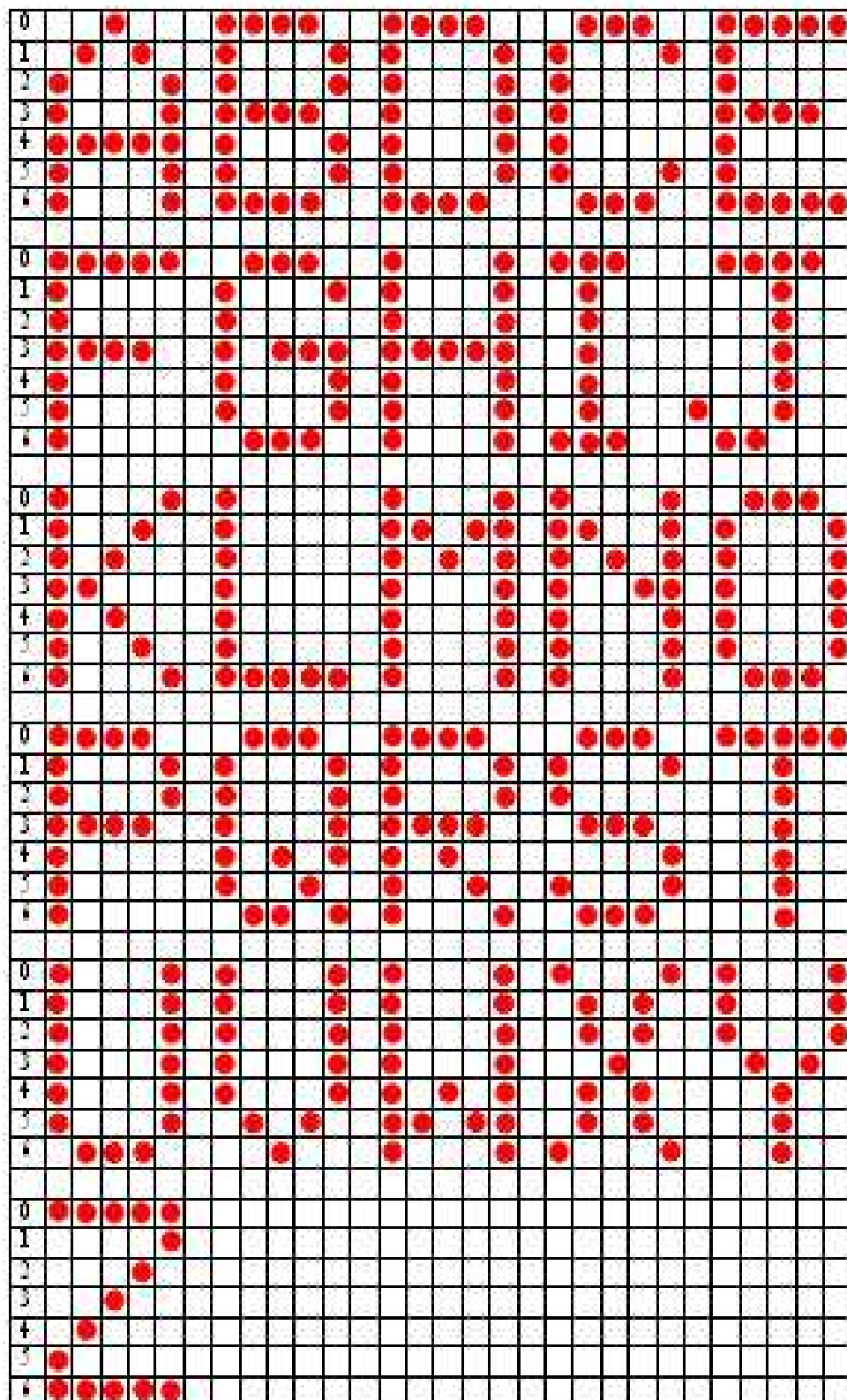
4.1. Hasil

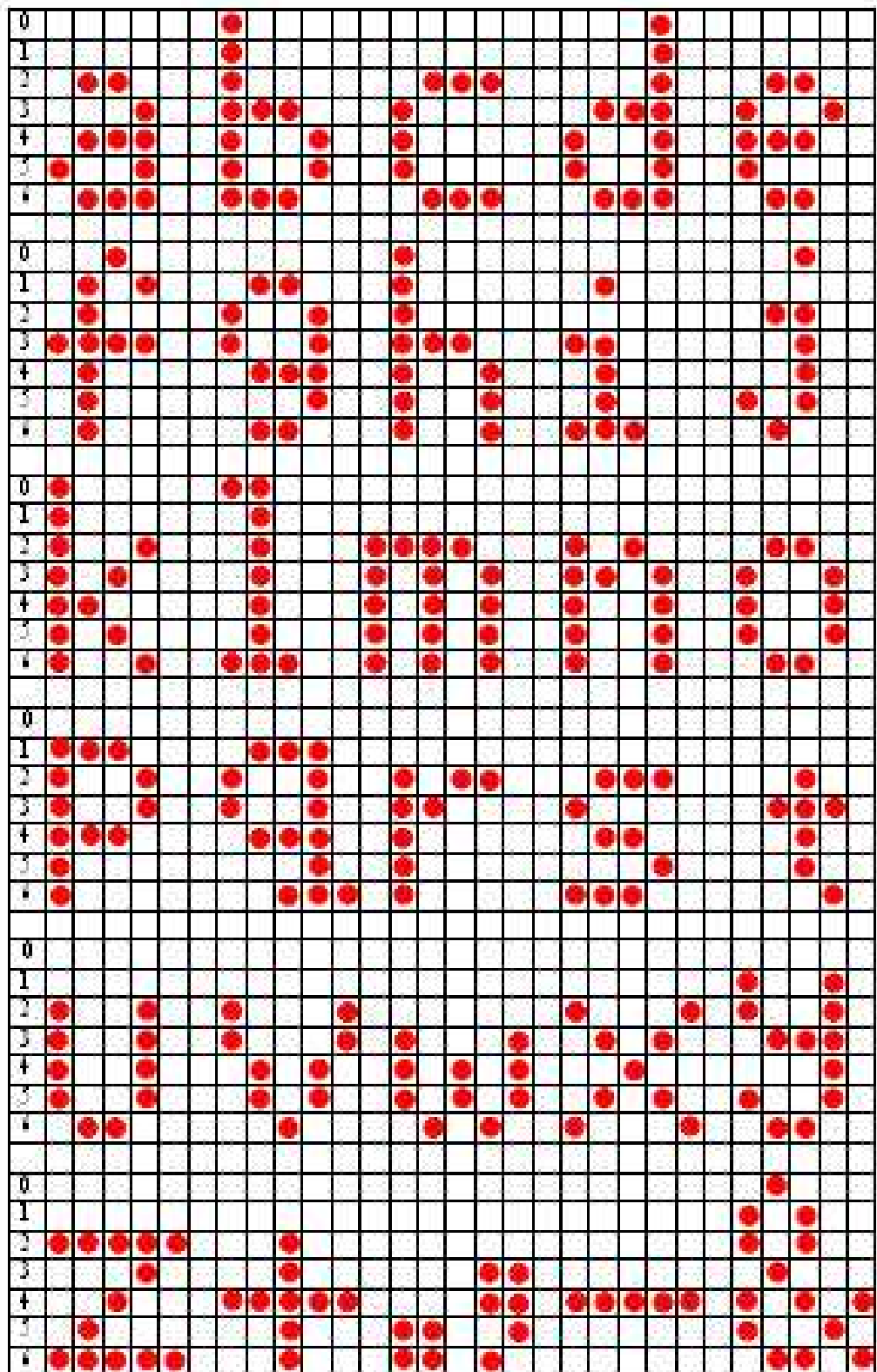
4.1.1. Tampilan Simulasi *Seven Segment*

Pada gambar dibawah merupakan contoh simulasi angka dan huruf yang akan ditampilkan pada rangkaian *seven segment* jika mengalami kendala seperti led mati atau gangguan pada kabel penghubung *port* sehingga menyebabkan angka dan huruf yang kita lihat menjadi kurang jelas.



Gambar 4.1 contoh tampilan simulasi angka





Gambar 4.2 contoh tampilan simulasi huruf

4.1.2. Pengujian Rangkaian *Seven Segment*

Pengujian selanjutnya adalah pengujian rangkain *seven segment* yang berfungsi untuk menampilkan *output* berupa angka dan huruf, pengujian rangkaian *seven segment* dapat dilihat pada gambar dibawah ini.

Untuk menampilkan *output* angka dua digit nol-nol pada rangkaian *seven segment* programnya adalah sebagai berikut:

```
!Kode0:
Kolom1 = &B00111110
Kolom2 = &B01000101
Kolom3 = &B01001001
Kolom4 = &B01010001
Kolom5 = &B00111110
RET
!Kode0:
Kolom1 = &B00111110
Kolom2 = &B01000101
Kolom3 = &B01001001
Kolom4 = &B01010001
Kolom5 = &B00111110
RET
```



Gambar 4.3 rangkaian *seven segment* angka nol-nol

Pengujian selanjutnya adalah pengujian rangkaian *seven segment* dengan *output* nol-satu. Untuk menampilkan *output* angka nol-satu pada rangkaian *seven segment* ini programnya adalah sebagai berikut:


```

!Kode0:
Kolom1 = &B00111110
Kolom2 = &B01000101
Kolom3 = &B01001001
Kolom4 = &B01010001
Kolom5 = &B00111110
RET

!Kode1:
Kolom1 = &B01000010
Kolom2 = &B01000001
Kolom3 = &B01111111
Kolom4 = &B01000000
Kolom5 = &B01000000
RET

```



Gambar 4.4 rangkaian *seven segment* angka nol-satu

Pengujian selanjutnya adalah pengujian rangkaian *seven segment* dengan *output* nol-dua. Untuk menampilkan *output* angka nol-dua pada rangkaian *seven segment* ini programnya adalah sebagai berikut:

```

!Kode0:
Kolom1 = &B00111110
Kolom2 = &B01000101
Kolom3 = &B01001001
Kolom4 = &B01010001
Kolom5 = &B00111110
RET

```



```

!Kode2:
Kolom1 = &B01000010
Kolom2 = &B01100001
Kolom3 = &B01010001
Kolom4 = &B01001001
Kolom5 = &B01000110
RET

```



Gambar 4.5 rangkaian *seven segment* angka nol-dua

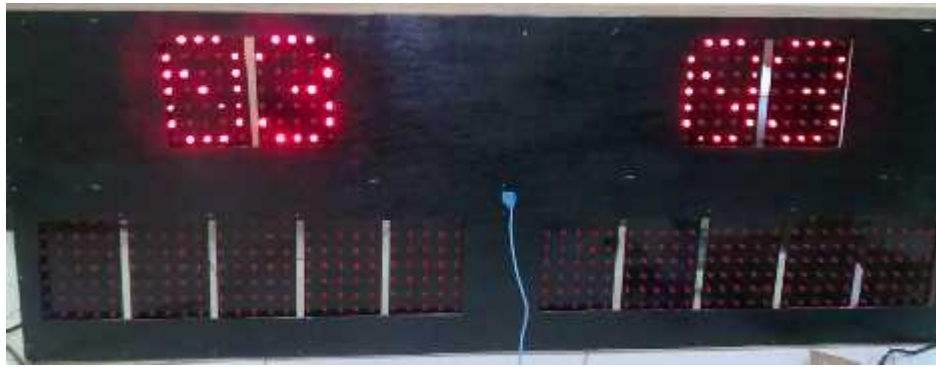
Pengujian selanjutnya adalah pengujian rangkaian *seven segment* dengan *output* nol-tiga. Untuk menampilkan *output* angka nol-tiga pada rangkaian *seven segment* ini programnya adalah sebagai berikut:

```

!Kode0:
Kolom1 = &B00111110
Kolom2 = &B01000101
Kolom3 = &B01001001
Kolom4 = &B01010001
Kolom5 = &B00111110
RET

!Kode3:
Kolom1 = &B00100010
Kolom2 = &B01000001
Kolom3 = &B01001001
Kolom4 = &B01001001
Kolom5 = &B00110110
RET

```



Gambar 4.6 rangkaian *seven segment* angka nol-tiga

Pengujian selanjutnya adalah pengujian rangkaian *seven segment* dengan *output* nol-empat. Untuk menampilkan *output* angka nol-empat pada rangkaian *seven segment* ini programnya adalah sebagai berikut:

```
!Kode0:
Kolom1 = &B00111110
Kolom2 = &B01000101
Kolom3 = &B01001001
Kolom4 = &B01010001
Kolom5 = &B00111110
RET

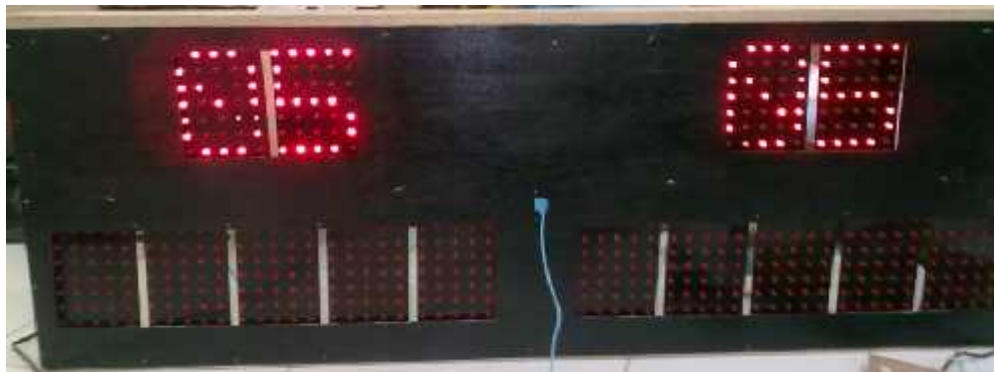
!Kode4:
Kolom1 = &B00111000
Kolom2 = &B00100100
Kolom3 = &B00100010
Kolom4 = &B01111111
Kolom5 = &B00100000
RET
```



Gambar 4.7 rangkaian *seven segment* angka nol-empat

Pengujian selanjutnya adalah pengujian rangkaian *seven segment* dengan *output* nol-lima. Untuk menampilkan *output* angka nol-lima pada rangkaian *seven segment* ini programnya adalah sebagai berikut:

```
!Kode0:
Kolom1 = &B00111110
Kolom2 = &B01000101
Kolom3 = &B01001001
Kolom4 = &B01010001
Kolom5 = &B00111110
RET
!Kode5:
Kolom1 = &B01001111
Kolom2 = &B01001001
Kolom3 = &B01001001
Kolom4 = &B01001001
Kolom5 = &B00110001
RET
```



Gambar 4.8 rangkaian *seven segment* angka nol-lima

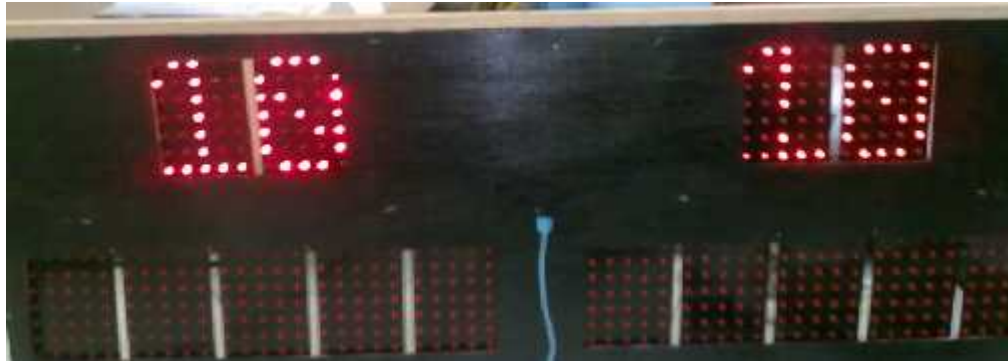
Pengujian selanjutnya adalah pengujian rangkaian *seven segment* dengan *output* satu-nol. Untuk menampilkan *output* angka satu-nol pada rangkaian *seven segment* ini programnya adalah sebagai berikut:

```
!Kode1:
Kolom1 = &B01000010
Kolom2 = &B01000001
Kolom3 = &B01111111
Kolom4 = &B01000000
Kolom5 = &B01000000
RET
```

```

!Kode0:
Kolom1 = &B00111110
Kolom2 = &B01000101
Kolom3 = &B01001001
Kolom4 = &B01010001
Kolom5 = &B00111110
RET

```



Gambar 4.9 rangkaian *seven segment* angka satu - nol

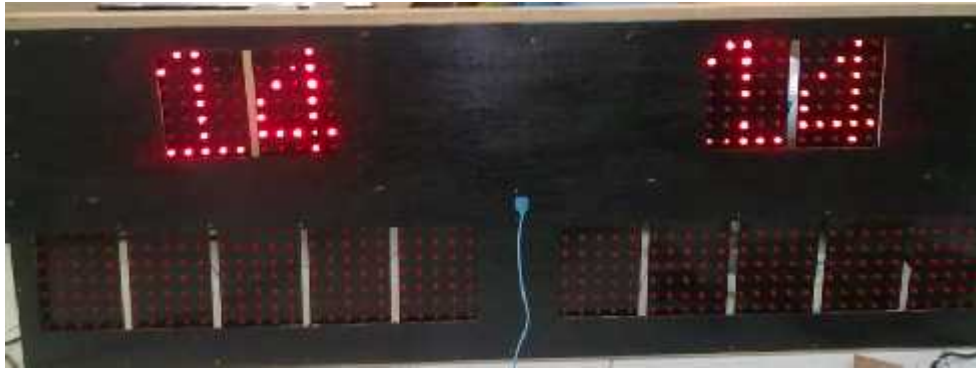
Pengujian selanjutnya adalah pengujian rangkaian *seven segment* dengan *output* satu-empat. Untuk menampilkan *output* angka satu-empat pada rangkaian *seven segment* ini programnya adalah sebagai berikut:

```

!Kode1:
Kolom1 = &B01000010
Kolom2 = &B01000001
Kolom3 = &B01111111
Kolom4 = &B01000000
Kolom5 = &B01000000
RET

!Kode4:
Kolom1 = &B00111000
Kolom2 = &B00100100
Kolom3 = &B00100010
Kolom4 = &B01111111
Kolom5 = &B00100000
RET

```



Gambar 4.10 rangkaian *seven segment* angka satu-empat

Pengujian selanjutnya adalah pengujian rangkaian *seven segment* dengan *output* satu-lima. Untuk menampilkan *output* angka satu-lima pada rangkaian *seven segment* ini programnya adalah sebagai berikut:

```
!Kode1:
Kolom1 = &B01000010
Kolom2 = &B01000001
Kolom3 = &B01111111
Kolom4 = &B01000000
Kolom5 = &B01000000
RET

!Kode5:
Kolom1 = &B01001111
Kolom2 = &B01001001
Kolom3 = &B01001001
Kolom4 = &B01001001
Kolom5 = &B00110001
RET
```



Gambar 4.11 rangkaian *seven segment* angka satu-lima

Pengujian selanjutnya adalah pengujian rangkaian *seven segment* dengan *output* dua-dua. Untuk menampilkan *output* angka dua-dua pada rangakian *seven segment* ini programnya adalah sebagai berikut:

```
!Kode2:
Kolom1 = &B01000010
Kolom2 = &B01100001
Kolom3 = &B01010001
Kolom4 = &B01001001
Kolom5 = &B01000110
RET
!Kode2:
Kolom1 = &B01000010
Kolom2 = &B01100001
Kolom3 = &B01010001
Kolom4 = &B01001001
Kolom5 = &B01000110
RET
```



Gambar 4.12 rangkaian *seven segment* angka dua-dua

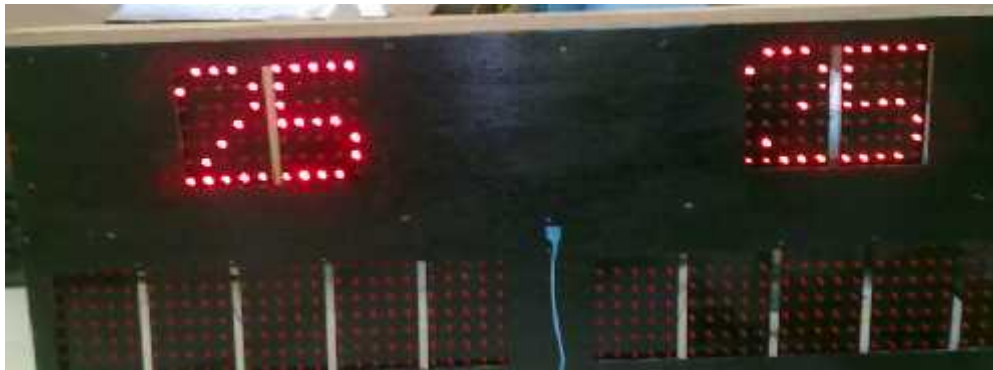
Pengujian selanjutnya adalah pengujian rangkaian *seven segment* dengan *output* dua-dua. Untuk menampilkan *output* angka dua-dua pada rangakian *seven segment* ini programnya adalah sebagai berikut:

```
!Kode2:
Kolom1 = &B01000010
Kolom2 = &B01100001
Kolom3 = &B01010001
Kolom4 = &B01001001
Kolom5 = &B01000110
RET
```

```

!Kode5:
Kolom1 = &B01001111
Kolom2 = &B01001001
Kolom3 = &B01001001
Kolom4 = &B01001001
Kolom5 = &B00110001
RET

```



Gambar 4.13 rangkaian *seven segment* angka dua-lima

Pengujian selanjutnya adalah pengujian rangkaian *seven segment* dengan *output* huruf, adapun program serta bentuk huruf dapat dilihat pada gambar berikut:

```

!Kodeab:
Kolom1 = &B01111110
Kolom2 = &B00010001
Kolom3 = &B00010001
Kolom4 = &B00010001
Kolom5 = &B01111110
RET

```



Gambar 4.14 rangkaian *seven segment* huruf A


```

!Kodebb:
Kolom1 = &B01111111
Kolom2 = &B01001001
Kolom3 = &B01001001
Kolom4 = &B01001001
Kolom5 = &B00110110
RET

```



Gambar 4.15 rangkaian *seven segment* huruf B

```

!Kodecb:
Kolom1 = &B00111110
Kolom2 = &B01000001
Kolom3 = &B01000001
Kolom4 = &B01000001
Kolom5 = &B00100010
RET

```



Gambar 4.16 rangkaian *seven segment* huruf C


```

!Kodedb:
Kolom1 = &B01111111
Kolom2 = &B01000001
Kolom3 = &B01000001
Kolom4 = &B00100010
Kolom5 = &B00011100
RET

```



Gambar 4.17 rangkaian *seven segment* huruf D

```

!Kodeab:
Kolom1 = &B01111110
Kolom2 = &B00010001
Kolom3 = &B00010001
Kolom4 = &B00010001
Kolom5 = &B01111110
RET

```

```

!Kodebb:
Kolom1 = &B01111111
Kolom2 = &B01001001
Kolom3 = &B01001001
Kolom4 = &B01001001
Kolom5 = &B00110110
RET

```

```

!Kodecb:
Kolom1 = &B00111110
Kolom2 = &B01000001
Kolom3 = &B01000001
Kolom4 = &B01000001
Kolom5 = &B00100010
RET

```

```

!Kodedb:
Kolom1 = &B01111111
Kolom2 = &B01000001
Kolom3 = &B01000001
Kolom4 = &B00100010
Kolom5 = &B00011100
RET

!Kodeeb:
Kolom1 = &B01111111
Kolom2 = &B01001001
Kolom3 = &B01001001
Kolom4 = &B01000001
Kolom5 = &B01000001
RET

```



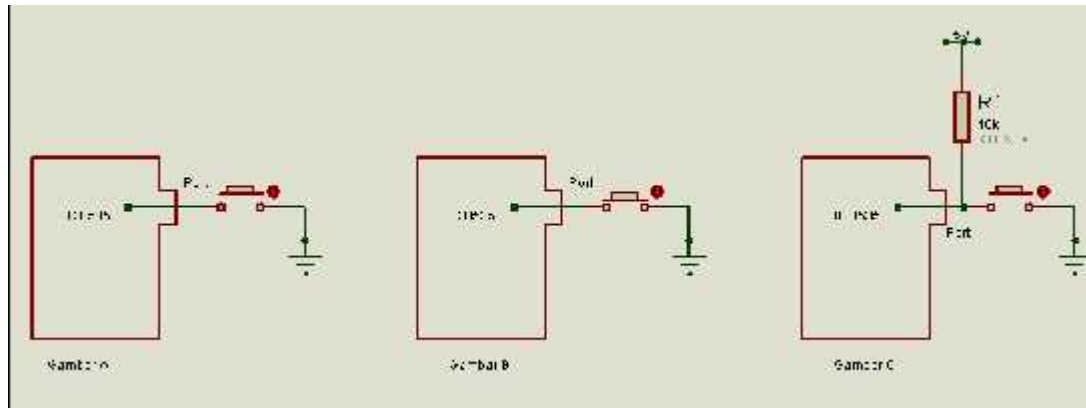
Gambar 4.18 rangkaian *seven segment* huruf A, B, C, D ,E

4.2. Analisis

4.2.1 Rangkaian Penstabil Sinyal

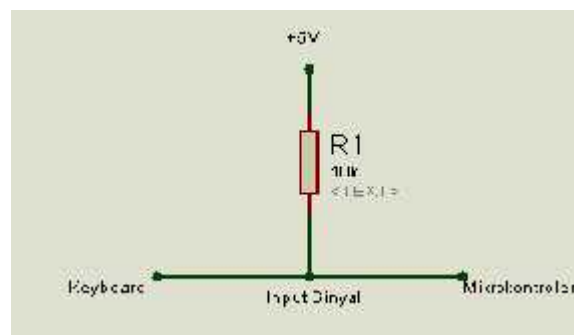
Pada IC mikrokontroler ATmega 8535 sebenarnya telah memiliki rangkaian *internal pull-up* yang berfungsi untuk membantu memberi logika *high* atau *low port* mikrokontroler. Prinsip kerja rangkaian *internal pull-up* dapat kita lihat pada gambar 4.19 dimana apabila *switch* pada gambar A tidak terhubung maka *port* mikrokontroler akan berlogika *x* sedangkan jika *switch* pada gambar B terhubung maka mikrokontroler akan berlogika *low*. Pada gambar C dimana

keluaran *port* dihubungkan ke resistor 10K (*pull-up*) kemudian diberi VCCin 5V maka *output* akan berlogika *high*.

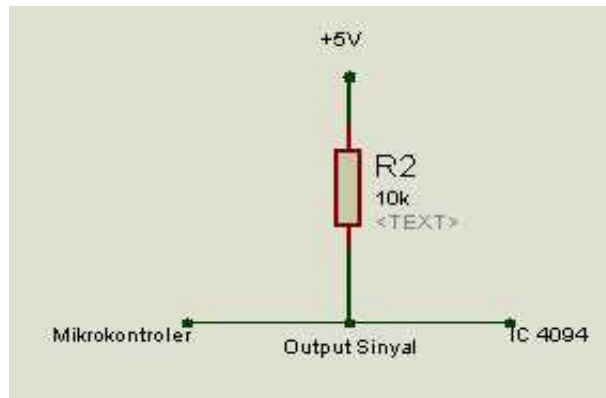


Gambar 4.19 rangkaian *pull-up* IC 8535

Setelah dilakukan pengujian pada perangkat keras terdapat permasalahan dimana data yang dikirim dari *keyboard* ke mikrokontroler dan data yang dikirim dari mikrokontroler ke IC 4094 mengalami gangguan sehingga menyebabkan tampilan menjadi tidak stabil, setelah dilakukan pengukuran terhadap pin mikrokontroler didapat tegangan bernilai 4V. Karena operasional mikrokontroler dan IC 4094 membutuhkan operasional pada kondisi *high* (1) atau 5V DC maka dibutuhkan rangkaian *pull-up* untuk penstabil sinyal seperti gambar 4.20 atau 4.21. Dari gambar 4.20 dapat dianalisa apabila data dari *keyboard* berlogika *high* (1) tetapi bernilai kurang dari 5V maka VCCin akan mensuplay tegangan yang masuk kemikrokontroler bila data yang dikirim adalah logika *low* (0) maka pin mikrokontroler akan berfungsi sebagai *ground*, begitu juga analisa kerja dari rangkaian 4.20.



Gambar 4.20 rangkaian penstabil *input* sinyal

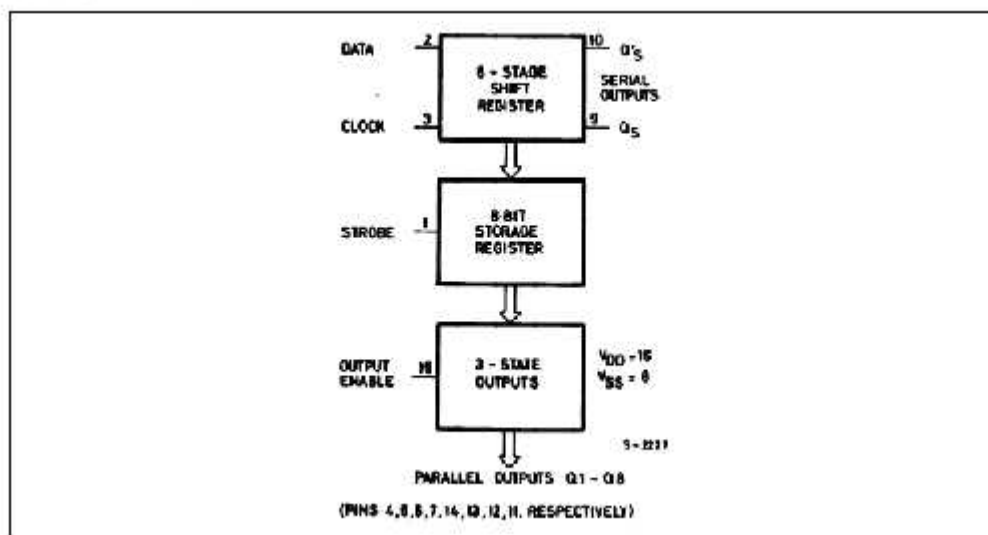


Gambar 4.21 rangkaian penstabil *output* sinyal

4.2.2 Proses Pengiriman Data Rangkaian *Seven Segment*

Agar membantu memudahkan kita memahami proses pengiriman data dan *clock* pada IC 4094 sebaiknya kita melihat terlebih dahulu diagram fungsional IC 4094 tersebut. Pada IC 4094 data dan *clock* yang masuk berupa *input* serial selanjutnya *strobe* dan *output enable* diberi logika *high* maka *output* yang dihasilkan berupa *output* paralel (Q1-Q8) sedangkan kaki Qs berfungsi untuk penyambungan IC yang sama, *input* data dan *clock* yang berada pada IC pertama akan digeser ke IC yang kedua melalui kaki Qs tersebut.







FUNCTIONAL DIAGRAM



Gambar 4.22 diagram fungsional IC 4094

Pada tabel baris ketiga menunjukkan salah satu sifat dari IC 4094 bila mana *clock* berlogika *high*, *output enable* berlogika *high*, *strobe* berlogika *low*, data sama dengan X (*don't care*) maka *output* paralel (Q1 dan Qn) yang dihasilkan tetap (*no change*).

TRUTH TABLE

CLOCK	OUTPUTS ENABLE	STROBE	DATA	PARALLEL OUTPUTS		SERIAL OUTPUTS	
				Q ₁	Q _n	Q _{7s}	Q _{6s}
	L	X	X	OC	OC	Q7	No Change
	L	X	X	OC	OC	No Change	Q7
	H	L	X	No Change	No Change	Q7	No Change
	H	H	L	L	Q ₁ - 1	Q7	No Change
	H	H	H	H	Q ₁ - 1	Q7	No Change
	H	H	H	No Change	No Change	No Change	Q7

X: Don't Care

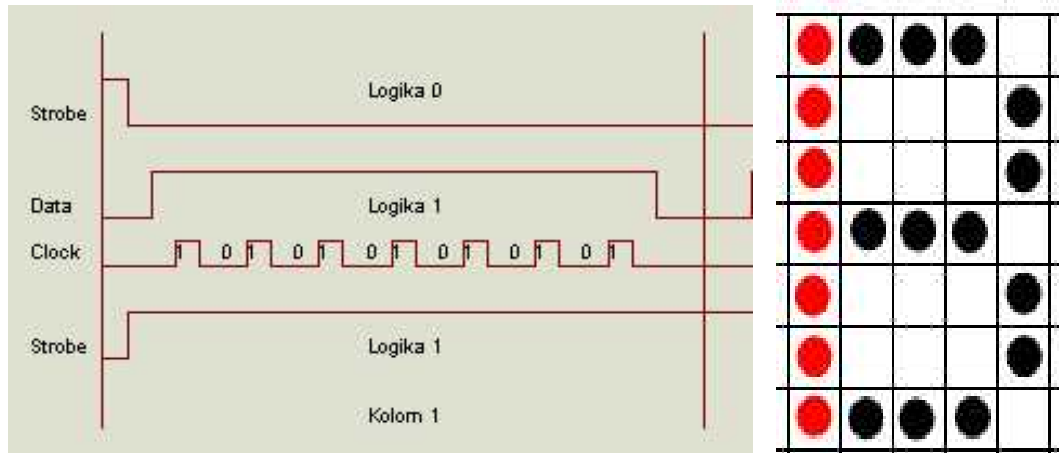
OC: Open Circuit

* At the positive clock edge information on the 7th shift register stage is transferred to the 6th register stage and the Q₆ output.

Proses pengiriman data pada rangkaian *seven segment* hingga membentuk satu karakter (contoh huruf B) meliputi beberapa tahap diantaranya adalah sebagai berikut OE (*output enable*) sama dengan +5V, *Strobe* dalam kondisi logika nol kemudian data pertama dikirim *clock* dalam kondisi logika nol, data kedua disiapkan *clock* diberi logika satu kemudian *clock* diberi logika nol lagi baru data kedua masuk, *clock* diberi logika nol pada saat akan memasukan data ketiga proses ini akan berulang hingga komplit mencapai 7 bit. Data pada kolom satu akan bergeser kekolom kedua begitu seterusnya sampai kekolom lima selanjutnya *strobe* diberi logika satu dan data akan disimpan dan kunci oleh IC 4094 sehingga mikrokontroler ATmega 8535 tidak akan *refresh* secara terus-menerus.

$S = 0$

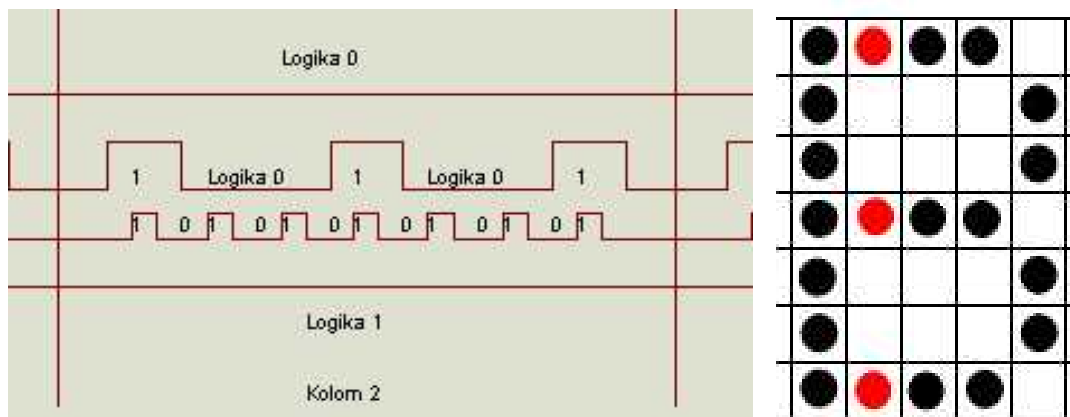
Shift out Data, Clock, & B01111111



Gambar 4.23 proses pengiriman data kolom 1

$S = 0$

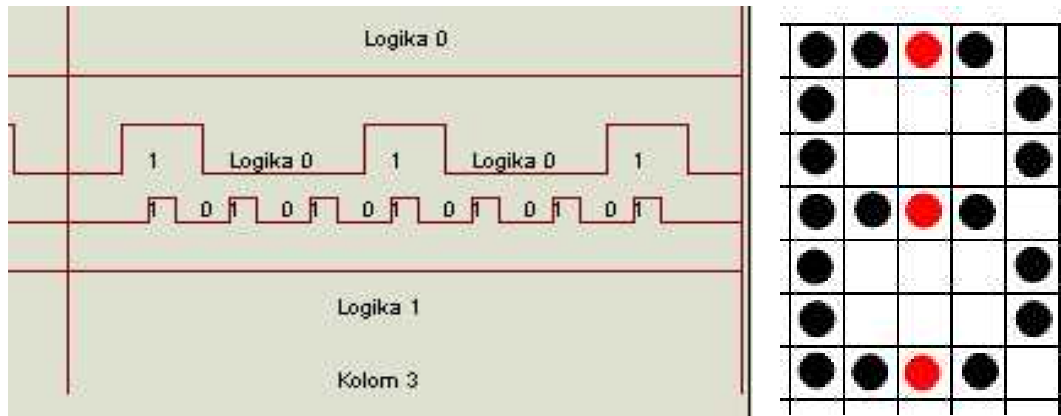
Shift out Data, Clock, & B01001001



Gambar 4.24 proses pengiriman data kolom 2

S = 0

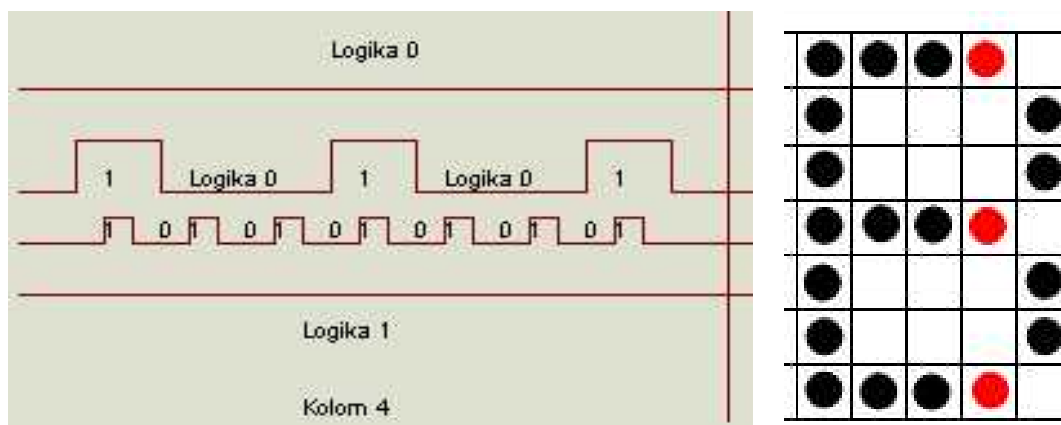
Shift out Data, Clock, &B01001001



Gambar 4.25 proses pengiriman data kolom 3

S = 0

Shift out Data, Clock, &B01001001

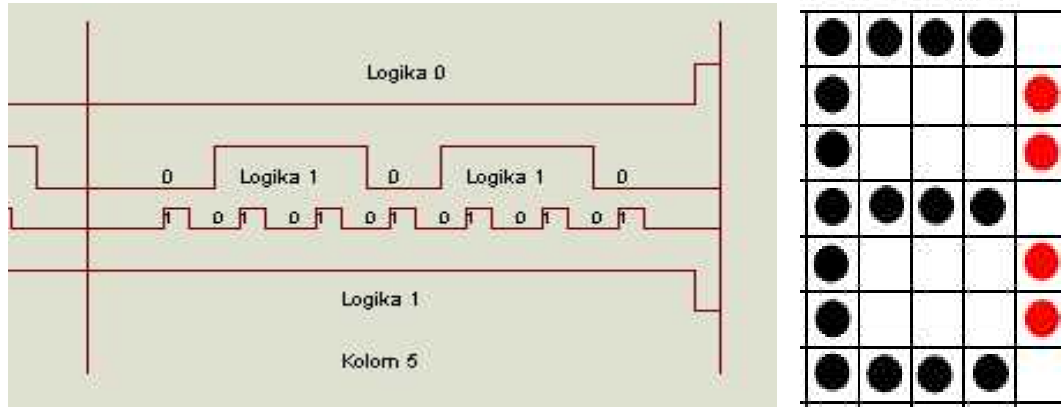


Gambar 4.26 proses pengiriman data kolom 4

S = 0

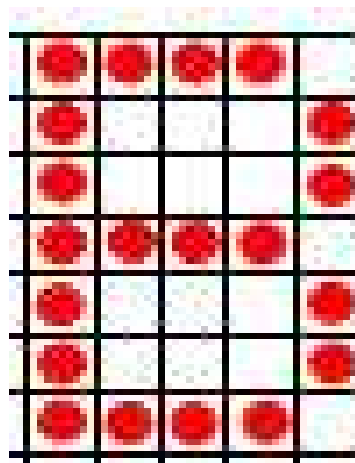
Shift out Data, Clock, & B00110110

S = 1



Gambar 4.27 proses pengiriman data kolom 5

Setelah melalui proses pengiriman data dari kolom satu sampai lima kemudian *strobe* diberi logika satu maka led akan menyala membentuk karakter huruf B seperti yang terlihat pada gambar 4.26 dan gambar pada kolom satu sampai lima saling berhubungan.



Gambar 4.28 karakter huruf B

Dengan menggunakan *keyboard* komputer sebagai *input* data, alat ini telah memiliki fitur dengan kemampuan menampilkan *output* berupa huruf dan angka. Proses pengoperasian alat yaitu :

- Untuk menambah skor regu 1 maka kita menekan tombol F5 pada *keyboard* selanjutnya menekan tombol angka 1
 - F5 → 1 = skor regu 1 bertambah sebanyak 1
 - F5 → 2 = skor regu 1 bertambah sebanyak 2
 - F5 → 3 = skor regu 1 bertambah sebanyak 3
- Untuk menambah skor regu 2 maka kita menekan tombol F6 pada *keyboard* selanjutnya menekan tombol angka 1
 - F6 → 1 = skor regu 2 bertambah sebanyak 1
 - F6 → 2 = skor regu 2 bertambah sebanyak 2
 - F6 → 3 = skor regu 2 bertambah sebanyak 3
- Untuk menghapus skor baik regu 1 maupun regu 2 kita menekan tombol F2 pada *keybord*
- Tombol Esc pada *keyboard* berfungsi sebagai *cancel* tambah skor untuk regu 1 maupun regu 2
- untuk mengisi nama regu 1 diawali dengan menekan tombol F3 kemudian kita mengetik nama regu yang akan bertanding
 - F3 → huruf a sampai z = nama regu dalam format huruf kecil
 - F3 → *Shift* + huruf A sampai Z = nama regu dalam format huruf besar
- untuk mengisi nama regu 2 diawali dengan menekan tombol F4 kemudian kita mengetik nama regu yang akan bertanding
 - F4 → huruf a sampai z = nama regu dalam format huruf kecil
 - F4 → *Shift* + huruf A sampai Z = nama regu dalam format huruf besar
- F1 berfungsi untuk menghapus semua display yang tampil pada papan skor (skor regu dan nama regu)

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari tahap pembahasan hasil dan analisa sistem papan skor olahraga berbasis mikrokontroler atmega 8535 dapat diambil kesimpulan sebagai berikut:

- 1) Pada rangkaian *display* papan skor olahraga, alat telah berhasil menampilkan *output* berupa karakter huruf dan angka .
- 2) Menggunakan *input keyboard* komputer sehingga fitur yang ditampilkan pada *display* lebih lengkap.

5.2 Saran

- 1) Pada rangkaian mikrokontroler dan antar muka sebaiknya digunakan catu daya yang ada sistem *current boost* menggunakan transistor daya.
- 2) Rangkaian catu daya yang menggunakan IC regulator akan lebih baik kinerjanya dengan menambahkan transistor daya.

DAFTAR PUSTAKA

Budi, Mayan. *Score Board yang Berbasis Mikrokontroler AT89S52*, Politeknik Sriwijaya Palembang, 2008.

Iswanto. *Design dan Implementasi Sistem Embedded Mikrokontroller ATmega8535 dengan Bahasa Basic*. Gava Media, Yogyakarta 2008.

Suryo, Kabut. *Pengendali Jarak Jauh Papan Pencatat Nilai Digital Olah Raga Karate*, Universitas Islam Indonesia, 2008.

Wahyudi, Didid Eka. *Papan Penampil Skor Pertandingan Tenis Meja Berbasis Mikrokontroler AT89C51*, Universitas Negeri Yogyakarta, 2010.

Wardhana, Lingga. *Belajar Sendiri Mikrokontroler AVR Seri ATmega 8535 Simulasi, Hardware, dan Aplikasi*, Andi, Yogyakarta, 2006.

http://en.wikipedia.org/wiki/Universal_Serial_Bus. diakses tanggal 14 maret 2011

<http://www.asciitable.com> diakses tanggal 26 april 2011